

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет прикладної математики

Кафедра системного програмування і спеціалізованих комп'ютерних систем

«До захисту допущено»

Завідувач кафедри

_____ В.П. Тарасенко

« ____ » _____ 2019 р.

Дипломна робота

на здобуття ступеня бакалавра

з напрямку підготовки 6.050102 «Комп'ютерна інженерія»

на тему: «Програмне забезпечення задач аналізу механіки деформованого тіла»

Виконав: студент IV курсу, групи КВ-53

Скілков Нікіта Володимирович _____

Керівник: асистент кафедри СПіСКС

Щербина Б.О. _____

Консультант з нормоконтролю: доцент кафедри СПіСКС,

к.т.н., доцент, Клятченко Я.М. _____

Рецензент: професор каф. ОТ, д.т.н., проф.,

Кулаков Ю.О. _____

Засвідчую, що у цій дипломній роботі
немає запозичень з праць інших авторів
без відповідних посилань.

Студент _____

Київ – 2019 року

**Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”**

Факультет прикладної математики

Кафедра системного програмування та спеціалізованих комп'ютерних систем

Освітньо-кваліфікаційний рівень “Бакалавр”

Напрямок підготовки 6.050102 “Комп'ютерна інженерія”

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ В.П. Тарасенко

“___” _____ 2019 р.

**З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЕКТ СТУДЕНТУ**

Скілкову Никіті Володимировичу

1. Тема проекту «ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ЗАДАЧ АНАЛІЗУ МЕХАНІКИ ДЕФОРМОВАНОГО ТІЛА»,
керівник проекту Щербина Богдан Олександрович, асистент каф.,
затверджені наказом по університету від “___” травня 2019 року № _____
2. Строк подання студентом проекту: “___” _____ 2019 р.
3. Вихідні дані для дипломного проектування: див. Технічне завдання.
4. Перелік задач, які потрібно вирішити:
 - розробити загальну структуру програми для задач аналізу механіки деформованого тіла;
 - виконати програмну реалізацію методу Рунге-Кутта;
 - виконати програмну реалізацію методу стрільби;
 - виконати програмну реалізацію методу скінченних елементів;
 - протестувати програмне забезпечення.
5. Перелік обов'язкового ілюстративного матеріалу:
 - цикл роботи системи (креслення);
 - алгоритм роботи методу Рунге-Кутта;
 - алгоритм методу стрільби;
 - алгоритм методу скінченних елементів.

6. Консультанти:

Питання	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Клятченко Я.М., доц., к.т.н.		

7. Дата видачі завдання: “___” _____ 2019 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломного проекту	Строк виконання етапів	Примітка
1.	Вивчення літератури за тематикою проекту	10.10.2018	
2.	Розробка та узгодження технічного завдання	29.10.2018	
3.	Аналіз існуючих рішень	5.12.2018	
4.	Створення структури програми	17.02.2019	
5.	Вибір середовища розробки та мови	20.03.2019	
6.	Розробка методу Рунге-Кутта	30.03.2019	
7.	Розробка методу стрільби	15.04.2019	
8.	Розробка методу скінченних елементів	30.04.2019	
9.	Відлагодження програмного продукту	1.05.2019	
10.	Підготовка пояснювальної записки	10.05.2019	
11.	Оформлення матеріалів проекту	20.05.2019	
12.	Попередній показ дипломного проекту на кафедрі	28.05.2019	

Студент _____ Скілков Н.В.

Керівник проекту _____ Щербина Б.О.

АНОТАЦІЯ

Кваліфікаційна робота включає пояснювальну записку (___ с., ___ рис., ___ табл., ___ додатки).

Об'єкт розробки - програмне забезпечення для вирішення задач аналізу механіки деформованого тіла. Воно дозволяє вирішити задачі метою яких є моделювання поведінки реальних фізичних тіл при вимушеній деформації. Завдяки цьому програмному забезпеченню можливо вирахувати характеристики твердого тіла які будуть вказувати на ступінь його деформації при заданому навантаженні.

Дане програмне забезпечення складається з реалізацій трьох методів:

- реалізації методу Рунге-Кутта, який послуговує для вирішення задач Коші для звичайних диференціальних рівнянь.
- реалізації методу стрільби, який потрібен для зведення крайової задачі до деякої задачі Коші для тої ж самої системи диференціальних рівнянь.
- реалізації методу дискретної ортогоналізації для вирішення одновимірної крайової задачі.

Програма вирішує задачу аналізу механіки твердого деформованого тіла. На практиці дозволяє отримати значення, що дозволяють зрозуміти ступінь деформації тіла при впливі на нього деякого навантаження.

Ключові слова: деформація, тверде тіло, метод Рунге-Кутта, метод стрільби, метод дискретної ортогоналізації, крайова задача, задача Коші.

ABSTRACT

The qualifying work includes an explanatory note (____ p., ____ figures, ____ tables, ____ annexes).

Object of development - software for solving tasks of analysis of mechanics of a deformed body. It allows to solve tasks, whose purpose is to simulate the behavior of real physical bodies with forced strain. This software allows to calculate the characteristics of the solid body which will indicate the degree of its deformation at a given load.

This software consists of implementation of three methods:

- Realization of the Runge-Kutta method, which serves to solve the Cauchy problem for ordinary differential equations.
- Implementation of the firing method required for the reduction of the boundary value problem to some Cauchy problem for the same system of differential equations.
- Implementation of the discrete orthogonalization method for solving a one-dimensional boundary value problem.

The program solves the problem of analyzing the mechanics of a solid deformed body. In practice, it allows you to obtain meaning that allows you to understand the degree of deformation of the body when exposed to some load.

Keywords: deformation, solid, Runge-Kutta method, shooting method, discrete orthogonalization method, boundary value problem, Cauchy problem.

[illegible]

[illegible]

[illegible]

ЗМІСТ

	стор.
1. НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ.....	2
2. ПІДСТАВИ ДЛЯ РОЗРОБКИ.....	2
3. МЕТА ВИКОНАННЯ.....	2
4. ДЖЕРЕЛА РОЗРОБКИ.....	2
5. ТЕХНІЧНІ ВИМОГИ.....	3
5.1 Вимоги до реалізації чисельних методів	3
5.2 Вимоги до апаратного забезпечення.....	3
5.3 Вимоги до програмного забезпечення.....	4
6. ЕТАПИ РОЗРОБКИ.....	4

					ІАЛЦ.045490.002 ТЗ			
Зм.	Арк.	№ докум.	Підп.	Дата				
Розроб.		Скілков Н.В.			Програмне забезпечення задач аналізу механіки деформованого тіла	Літ.	Аркуш	Аркушів
Перевір.		Щербина Б.О.					1	4
					Технічне завдання	КПІ ім. Ігоря Сікорського ФПМ КВ-53		
Н. контр.		Клятченко Я.М.						
Затв.		Тарасенко В.П.						

1. НАЙМЕНУВАННЯ І ОБЛАСТЬ ЗАСТОСУВАННЯ

Найменування роботи – «Програмне забезпечення задач деформованого тіла».

Область застосування: використання для вирішення задач аналізу твердого деформованого тіла

2. ПІДСТАВА ДЛЯ РОЗРОБКИ

Підставою для розробки є завдання на дипломне проектування, затверджене кафедрою системного програмування і спеціалізованих комп'ютерних систем Національного технічного університету України «Київський Політехнічний Інститут».

3. ЦІЛЬ І ПРИЗНАЧЕННЯ РОБОТИ

Метою даного проекту є створення програмного забезпечення для вирішення задач аналізу механіки твердого деформованого тіла.

4. ДЖЕРЕЛА РОБОТИ

Джерелами роботи є науково-технічна література з численних методів та фізики, статті та публікації взяті с технічних видань та мережі Інтернет.

					ІАЛЦ. 045490.002 ТЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		2

5. ТЕХНІЧНІ ВИМОГИ

5.1 Вимоги до реалізації чисельних методів:

- На вхід подається система диференційних рівнянь у вигляді матриці коефіцієнтів та умови задачі Коші.
- На основі вхідних даних розраховуються і подаються на вихід коефіцієнти рядів Тейлора.
- В разі виникнення помилок при розрахунку результату програма не видає результат і друкує помилку.
- Виходом є матриця коефіцієнтів.

5.2 Вимоги до апаратного забезпечення:

- Монітор.
- Комп'ютер.
- Мобільний пристрій.

5.3 Вимоги до програмного забезпечення:

- Операційна система Windows
- Visual Fortran

					ІАЛЦ. 045490.002 ТЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		3

6. ЕТАПИ РОЗРОБКИ

№ з/п	Назва етапів роботи та питань, які мають бути розроблені відповідно до завдання	Термін виконання
1.	Видача завдання на дипломне проектування	4.11.2018
2.	Вивчення літератури за тематикою проекту	10.10.2018
3.	Розробка та узгодження технічного завдання	29.10.2018
4.	Аналіз існуючих рішень	5.12.2018
5.	Створення структури програми	17.02.2019
6.	Вибір середовища розробки та мови	20.03.2019
7.	Розробка методу Рунге-Кутта	30.03.2019
8.	Розробка методу стрільби	15.04.2019
9.	Розробка методу скінченних елементів	30.04.2019
10.	Відлагодження програмного продукту	1.05.2019
11.	Підготовка пояснювальної записки	10.05.2019
12.	Оформлення матеріалів проекту	20.05.2019
13.	Попередній показ дипломного проекту на кафедрі	28.05.2019

[illegible]

Поз.	Формат	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кількість аркушів	№ прим.	Примітки
	A4	ІАЛЦ.045490.007 ДЗ	Програмне забезпечення	1		
			для задач аналізу			
			механіки твердого			
			деформованого тіла.			
			Алгоритм роботи			
			методу стрільби			
			Схема алгоритму			
	A4	ІАЛЦ.045490.008 Д4	Програмне забезпечення	1		
			для задач аналізу			
			механіки твердого			
			деформованого тіла.			
			Алгоритм роботи			
			методу скінченних			
			елементів			
			Схема алгоритму			
		CD-ROM	Матеріали бакалаврського	1		
			проекту			
ІАЛЦ.045490.003 ТП						Арк.
						2
Змін.	Арк.	№ докум.	Підпис	Дата		

ЗМІСТ

	стор.
ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ.....	4
ВСТУП.....	5
1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ, ОБҐРУНТУВАННЯ ТЕМИ ДИПЛОМУ.....	6
1.1 Загальні рішення, що існують сьогодні на ринку.....	6
1.2 Аналіз Femar – основного продукту на ринку.....	13
1.3 Обґрунтування теми дипломного проекту	17
2. ТЕХНОЛОГІЧНІ ТА ПРОГРАМНІ ЗАСОБИ РОЗРОБКИ.....	18
2.1 Класифікація ЧМ.....	18
2.2 Алгоритм методу Рунге-Кутта.....	19
2.3 Алгоритм методу стрільби.....	19
2.4 Алгоритм методу кінцевих елементів.....	21
2.5 Приклад реалізації методу Рунге-Кутта на мові Fortran.....	25
2.6 Приклад реалізації методу Рунге-Кутта на мові C/C++.....	27
2.7 Приклад реалізації методу Рунге-Кутта на мові Python.....	29
2.8 Порівняльна характеристика мов програмування для даної задачі.....	30
3. ПРОГРАМА ДЛЯ РОЗРАХУНКУ ЗАДАЧІ ТВЕРДОГО ДЕФОРМОВАНОГО ТІЛА.....	31
3.1 Загальна структура програми.....	31
3.2 Функціонування компонент програми.....	40
3.3 Аналіз результатів.....	42
ВИСНОВКИ.....	44
Список використаної літератури.....	47

					ІАЛЦ.045490.004 ПЗ			
Зм.	Лист	№ докум.	Підп.	Дата	Програмне забезпечення задач аналізу механіки деформованого тіла	Літ.	Аркуш	Аркушів
Розробив	Скілков Н.В.							
Перев.	Щербина Б.О.						1	50
						КПІ ім. Ігоря Сікорського, ФПМ, КВ-53		
Н. контр.	Клятенко Я.М.							
Затвер.	Тарасенко В.П.				Пояснювальна записка			

ДОДАТКИ

Додаток 1. Графічний матеріал

ІАЛЦ.045490.005 Д1. Цикл роботи системи. Схема структурна.

ІАЛЦ.045490.006 Д2. Алгоритм роботи методу Рунге-Кутта. Схема алгоритму.

ІАЛЦ.045490.007 Д3. Алгоритм роботи методу стрільби. Схема алгоритму.

ІАЛЦ.467200.008 Д4. Алгоритм роботи методу скінченних елементів. Схема алгоритму.

Додаток 2. Лістинг програми

Додаток 3. Презентація

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		2

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ

ПЗ – програмне забезпечення

ЧМ – чисельні методи

СЗДР – система звичайних диференційованих рівнянь

Задача Коші – це одна з основних задач теорії диференціальних рівнянь, рішенням якої є інтеграл що задовольняє початкові умови.

Початкові умови – множина умов що ставляться на диференційоване рівняння або СЗДР, приводить до виникнення задачі Коші.

Крайова задача – задача, що полягає в знаходженні рішення заданого диференціального рівняння або СЗДР, яке задовільнює крайовим (граничним) умовам в кінцях інтервалу або на границі області. Для гіперболічних та параболічних рівнянь вони часто мають також початкові умови, тому ї називають змішаними задачами.

Звичайне диференційне рівняння (ЗДВ) – диференційне рівняння для функції однієї змінної.

Диференційне рівняння (ДР) – рівняння яке складається з функції та її похідних. Коренем є сама функція яка задовольняє умови рівняння.

Порядок диференційного рівняння – максимальна ступінь похідної в рівнянні.

Інтерполяція – в обчислювальній математиці спосіб знаходження проміжних значень величини за наявним дискретним набором відомих значень.

Апроксимація - наближене вираження одних математичних об'єктів іншими, близькими за значенням, але простішими, наприклад, кривих ліній — ламаними, ірраціональних чисел — раціональними, неперервних функцій — многочленами.

Краштест – тестування будь-якого продукту на витримування пошкоджень.

Сплайн – функція, область визначення якої розбита на шматки, і на кожному функція являє собою поліном.

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		3

Система алгебраїчних рівнянь (САР) – система рівнянь, де кожне рівняння є алгебраїчним.

FEA-пакети – програми що засновані на методі кінцевих елементів (finite element analysis).

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		4

ВСТУП

За останній час чисельні методи стали використовуватися всюди. Складно знайти галузь виробництва, де б вони не фігурували. Вони дозволяють значно зекономити витрати підприємств при впровадженні, тестуванні або автоматизації будь-якого продукту. Також ЧМ дозволяють робити високоточні розрахунки для передових галузей науки.

Чисельні методи (ЧМ), розглянуті у даній роботі, дозволяють змодельовати і розрахувати вплив навантаження на тверде анізотропне тіло і його деформацію. Це дозволяє змодельовати ситуацію де тіло деформується замість проведення його реальної деформації і заощадити багато витрат і часу. Конкретним прикладом може слугувати концерн Deimler, який випускає всім відомий Mercedes. Для краштестів своїх автомобілів вони використовують спочатку моделювання, де в спеціальні програми заносять дані про геометрію кузова, матеріал, силу удару і інші дані, що потрібні для точного розрахунку. На жаль, принцип роботи програмного забезпечення (ПЗ), яке вони використовують, є комерційною таємницею, і тому принципи його роботи є закритою інформацією, але засади на яких воно ґрунтується відомі, і саме їх розглянуто у даній роботі.

Кожен з методів які проаналізовані у даній роботі достатньо відомі, і деякі детально розглянуті при вивченні дисципліни чисельні методи у київському політехнічному інституті. Зараз існує багато методів для вирішення задач аналізу механіки твердого тіла, але тут розглядаються найбільш поширені, а також достатньо прості, такі як метод Рунге-Кутта або метод стрільби. Для вирішення задач цього типу є і більш складні методи, такі як метод скінчених елементів (МСЕ), які використовують для складних задач. Але їх алгоритмічна складність, час роботи та складність реалізації є надлишковим для задач цього проекту. Але і методи, які використовуються в

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		5

даній роботі, є оптимальними, і мають достатньо низьку похибку для того, щоб використовуватися для серйозних задач.

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		6

1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ, ОБҐРУНТУВАННЯ ТЕМИ ДИПЛОМНОГО ПРОЕКТУ

1.1 Загальні рішення, що існують сьогодні на ринку

Задачі що вирішують FEA-пакети (finite analysis algorithm) на сьогоднішній день є різноманітними і охоплюють майже всі частини інженерних розрахунків: міцність, коливання та стійкість, різні задачі з акустики, динаміка, гідродинаміка, аеродинаміка тощо. Представити повний список задач до яких причетні програми засновані на базі методу скінченних елементів та йому подібних, як і повний список всіх пакетів програм, практично неможливо. В цій же роботі зроблено аналіз багатьох поширених пакетів. Вважається що одні FEA-пакети мають перевагу над іншими. Така думка складається в тому числі через те що всі поширені продукти мають закритий код, і тому точно невідомо як він працює. Також причиною такої думки є те що багато з тих хто використовує ПЗ знають лише один продукт, і ніколи не чули про існування інших. В вищих закладах також як правило використовують одне з рішень, тому спеціалісти не цікавляться всім кругом представлених FEA-пакетів на ринку. Виникають необґрунтовані міфи про переваги і областях застосування цих систем.

Безперечно, головним ПЗ в галузі FEA-програм є пакет ANSYS. Це видно з кількості зареєстрованих ліцензійних користувачів, яких понад мільйон, представлені широкі можливості і гнучкість налаштувань. Також про це говорить велика чимала кількість серйозних компаній, що використовують даний пакет, а також ціна, що є вищою за середню на ринку (проте з великою кількістю користувачів) та й одні з найвищих вимог до обладнання серед всіх FEA-пакетів (для версії 5.5.1 це комп'ютер з 256 Мбайт оперативної і 96 Мбайт відеопам'яті). Більш детальний опис пакета ANSYS можна знайти на офіційному сайті компанії ANSYS [5], також існує багато інструкцій та матеріалів по його використанню та функціоналу. Великим плюсом цієї

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		7

програми є доступна і повна документація, завдяки якій спеціаліст маючи глибокі знання в галузі зможе використовувати цю програму не звертаючись до служби підтримки або досвідчених користувачів. Також програма представляє велику кількість кінцевих елементів – більше 100, проте не є зрозумілим чи справді є необхідність в такій кількості і чи не є це надлишковим. Так, наприклад, є матеріал [6], де доводиться достатньо легким і швидким способом надлишковість таких кінцевих елементів, як «Сендвіч» (в ANSYS - Shell91), і те, що більшість спеціалістів, які використовують його, роблять це неправильно. Недоліком цього ПЗ можна назвати службу підтримки, як показує практика вона нечасто допомагає вирішити проблеми при використанні пакету і відповідає часто стандартними відповідями автовідповідача. Програма має зручний інтерфейс, особливо порівнюючи з іншими програмами даного класу, але значно поступається (майже в усіх основних аспектах, таких як простота та зрозумілість) стандартному інтерфейсу під всі основні пакети що були розроблені нещодавно - FEMAP (розробник Enterprise Software Products).

З ANSYS може зрівнятися по кількості установок та легальних ліцензійних користувачів пакет під торговою маркою NASTRAN (офіційно цією маркою володіє NASA (Cosmic)). Під цією маркою створені такі найпоширеніші комплекси ПЗ: MSC/NASTRAN (MacNeal-Schwendler Corporation) і UAI/NASTRAN (Universal Analytics). Дещо менш поширеними є NE/NASTRAN (Noran Engineering) і ME/NASTRAN (MacroIndustries). Не дивлячись на значно нижчу ціну пакетів UAI, MSC, NE, ці рішення нічим не гірші від комплексу ANSYS, а іноді мають кращий розвиток, функціонал та можливості. Наприклад, в цих пакетах вже значний час (більше 5 років) представлений «передубомовлений метод зв'язних градієнтів Холесського», в той час як в ANSYS його нема до 1999 року. Цей метод є оптимальним (найбільш точним і швидким) для вирішення матриць високого рангу. Недоліком пакетів NASTRAN порівняно з ПЗ ANSYS можна відмітити тільки

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		8

обмежену кількість операційних систем та платформ, що підтримуються. Проте в ці пакети гарно інтегровані FEMAP-оболонки, які працюють на більшості актуальних платформ, тому цей недолік не є вирішальним, оскільки достатньо нечасто використовуються платформи, що їх не підтримують, а саме ПЗ не потребує серйозних ресурсів на відміну від того самого пакету ANSYS. Наприклад, для версії пакету MSC/NASTRAN v70.5 мінімально необхідним процесором є 486DX, а також 16 Мбайт оперативної пам'яті і операційна система Windows 95 або вище. Інколи такі вимоги приводять до питань реальних можливостей такої програми, але якщо розглянути правильно розроблену систему, зрозуміло що при належній оптимізації алгоритми ЧМ, що закладені в основі не можуть потребувати великих ресурсів.

Також є два інших пакета, що набули популярності ще за часів кінця СРСР. Це таке ПЗ як Algor (колись на ринок потрапив з ім'ям Aldan) – продукт однойменної компанії Algor, а також пакет COSMOS/M – продукт компанії Structural Research and Analysis Corp. Ці пакети не мають такого значного поширення як попередні два. Забезпечення COSMOS/M використовувалось в оборонній промисловості, але згодом його витіснила програма UAI/NASTRAN, а Algor - пакет DesignSpace (який є по суті полегшеним варіантом комплексу ANSYS). Не зовсім зрозуміло чому пакет COSMOS/M не набув популярності, оскільки він завжди був доступним на актуальних платформах, та мав сучасні оновлення і інтерфейс. А ось втрата ринку Algor'ом цілком зрозуміла для всіх його користувачів. В останній 13-й, наприклад, версії ПЗ Algor, навіть попри новий інтерфейс SuperDrawIII, видно що базою залишилась стара восьмирозрядна DOS. Вікна для розрахунку, а також додаткової інформації залишилися в стилі MS-DOS v3.3, що робить їх достатньо незручними порівняно з сучасними інтерфейсами. З плюсів документація представляє, як точно Algor вирішує завдання на пластику, динаміку тощо, також показаний краштест скла на удар, наприклад. Значним недоліком для промислового використання є те ще офіційного поставника ПЗ

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		9

на вітчизняний ринок немає, а також служба підтримки цього продукту нечасто відповідає на питання, і ситуація в цілому схожа з пакетом ANSYS.

Незалежні тести всіх представлених комплексів ПЗ можна подивитись на цьому сервері [7], проте особливої різниці як в швидкості, так і в точності розрахунків там не має. Одна з небагатьох відмінностей – повільна генерація сіток в пакеті Algor. Усі інші параметри всіх ПЗ коливаються в діапазоні 10%. Раніше (до версії 5.5.x) ANSYS мав велику відмінність в розрахунку великих систем рівнянь (більше 100 000 рівнянь) порівняно з іншими проаналізованими комплексами ПЗ на більше ніж 15%. В цілому складно знайти великі відмінності в роботі цих програм, оскільки всі вони використовують один і той же клас ЧМ, які базуються методі скінченних елементів, методі Рунге-Кутта тощо. Математичний апарат ґрунтується на одних й тих же алгоритмах, тому очікувати того що одна з програм буде значно швидшою за іншу немає сенсу, розбіжність виникає за рахунок деталей реалізації і стеку технологій.

Проте, на ринку програм кінцево-елементного аналізу пострадянські виробники зробили один конкурентоспроможний продукт - пакет «Ліра». Цей комплекс ПЗ перевершував усі поширені аналоги по швидкодії, причому не поступаючись в точності розрахунків. Інше ПЗ, яке мало багато перспектив - це МАК, до 1990 року - Think (головний розробник Маслов А.А.), розробник - фірма «Унікон». По функціоналу набагато кращий за американський пакет STRAP для аналогічної області застосування, однак він теж не набув популярності. Проблема в цих продуктах була в слабкій документації, а також в поганому маркетингу, фактично, про ці комплекси ПЗ мало хто з спеціалістів знав коли вони вийшли.

Окрема група ПЗ - freeware- і shreeware-пакети. Вони хоча й вільні для використання, проте це не означає, що вони неточні або повільні. Як вже було сказано, всі FEA-програми базуються на спільних принципах, тому великої

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		10

різниці в їх роботі не може бути. Спеціалісти в цій галузі знають що використання вільного для розповсюдження програмного забезпечення не гірше від комерційних пакетів. В інших галузях є багато вільного програмного забезпечення: Internet-пакет Netscope Communicator, операційні системи Linux і freeBSD, Web-сервер Apache, офісний пакет StarOffice та LibreOffice, дизайнерський пакет CorelDRAW (for Linux) тощо. Але freeware-програми скінчено-елементного аналізу не так відомі, оскільки вони не є такими поширеними. Багато їх можна знайти на Web-сторінках [8]. Розглянемо деякі найбільш поширені з них.

DANFE - програма для аналізу напружень, зусиль, зсувів і деформацій в конструкціях за допомогою методу скінченних елементів (МСЕ). Програма може бути завантажена вже готовим до виконання файлом, а також проектом з вихідним кодом, що може бути зібраний на будь-якій системі, що має Fortran-компілятор.

Системні вимоги:

- 1,5 Мбайт на жорсткому диску для вихідного тексту і вихідного коду програми;
- 2-50 Мбайт дискового простору для даних;
- 5 Мбайт або більше оперативної пам'яті для 2D-завдань і до 80 Мбайт оперативної пам'яті для великих тривимірних нелінійних задач;
- операційна система: DOS, Windows, OS / 2, HP-UX, Solaris, Cray, Fujitsu VPX, IBM RS / 6000s.

Можливості:

- 18 різних типів елемента (в тому числі 4D з 32 вершинами і геометрією, що залежить від часу);

- шість оптимальних вирішальних пристроїв, що включають в себе розкладання за Гаусом і метод Холесського, непрямі, або ітераційні вирішальні пристрої, в тому числі предобумовлений метод зв'язних градієнтів, який в свою чергу оптимізований і для скалярних і векторних процесорів;
- всі матеріали можуть бути нелінійні (за фон Мізеса, Трісці, Мору-Кулону), в'язкопластичні, а також як із залишковими деформаціями, так і без них.

Програма на ринку вже більше 30 років, основна розробка ведеться в університеті Манчестера, Великобританія (University of Manchester UK), колективом дослідників, що очолює професор І.М.Сміт (I.M.Smith). Також в цій програмі представлені графічні пост/препроцесори для візуалізації.

Порівняно з комерційними, це FEA-пакет має багато переваг, таких як: представлені вихідні коди, тому можна зрозуміти як точно вона працює, підтримка багатопроцесорності (при інсталяції Linux на двухпроцесорній mainboard розрахунки проходять швидше, ніж комплекс ПЗ ANSYS на Sun-системах), наявність елементів із геометрією, що залежить від часу (ефекти плинності, самонапруження, релаксації без зовнішніх впливів), довгий час на ринку (більше, ніж ANSYS), авторитетний для всього світу виробник.

Також є FEA-пакет з назвою Mefisto, його розробником є лабораторія чисельного аналізу університету П'єра і Марії Кюрі (Париж, Франція), детальна інформація може бути знайдена на сайті розробника [10]. Існують версії для платформ: DEC, IBM, SUN, PC. Працює під UNIX-системами при наявності бібліотеки X11 (для Linux і FreeBSD - стандартно поставляється бібліотека). Вимоги: 40 Мбайт на жорсткому диску для виконуваних файлів, 128 Мбайт віртуальної (дискової) пам'яті для тимчасових файлів, 16 Мбайт оперативної пам'яті. На ринку представлений в двох варіантах:

- для конкретної системи і платформи готовою програмою;
- в вихідним кодом на Fortran77.

Для розуміння масштабу цієї програми є такий факт, що текст програми (остання версія з липня 1999 роки) містить 267 319 рядків (більше кілометра роздруківки на рулонної паперу при щільності 12 рядків на дюйм), з яких 117 132 рядки – коментарі. Серед документації подається: «Керівництво користувача по пост/препроцесору», «Керівництво користувача для вирішення задач механіки», «Керівництво користувача для вирішення задач теплодинаміки» тощо.

Розглянуті програми з правами вільного розповсюдження є точними, швидкими та зручними комплексами ПЗ, що базуються на методі скінченного елемента. Вони також є повністю гнучкими для конфігурації з необхідними для вас можливостями і можуть законно використовуватись для будь-яких проектів.

Перевагою некомерційних пакетів є допомога від розробників в будь-яких складних питаннях, проте на прості розробник скоріш за все не знайде часу відповідати, а документація зазвичай небагата. У комерційних продуктах, навпаки, служба підтримки з частіше за все процитує вам керівництво користувача для простого питання, але проігнорує складне питання. Також перевагою некомерційних програм є факт, що купуючи комерційний FEA-пакет, ви купуєте програму, яка невідомо конкретно як працює, і тільки від розробника залежить зможе вона правильно обрахувати задачу чи ні. У безкоштовних програмах, навпаки, чітко з вихідного коду видно як саме працює програма, і при необхідності можна змінити деякі ділянки.

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		13

1.2 Аналіз Femap – основного продукту на ринку

Перша мета Femap полягає в тому, щоб дозволити вам або створювати моделі з кінцевими елементами з нуля, або імпортувати і повторно використовувати файли CAD з різних форматів.

Хоча в деяких випадках це може бути простий випадок зчеплення, у багатьох випадках при дослідженні більш екзотичної форми моделювання, це буде включати абстрагування від необробленої геометрії. Це полягає у створенні більш відповідних або легких зображень частини або збірки - чи то зразкові моделі (наприклад, видалення невеликих об'єктів, філе, фаски та дрібні деталі) або побудова моделей середньої площини або балки.

Хоча Femap може похвалитися вражаючим діапазоном зрілих і добре розроблених інструментів, з кожним випуском завжди є місце для розширення. У випуску Femap 12, деякі ключові оновлення варто вивчити.

Перший - це робота стібка, щоб об'єднати поверхні в єдине тіло; це буде корисним для роботи з тими менш ідеальними моделями, де окремі поверхні не добре з'єднані.

Це тепер дозволяє задати максимальну толерантність і використовує ітераційний підхід для отримання зшивних поверхонь без надмірного клопоту або ручної переробки.

Також було проведено роботу по булевим операціям для кращої роботи із збереженням комбінованих кривих і граничних поверхонь, які зазвичай використовуються для зменшення складності геометрії для поліпшення якості сітки.

Існує також нова опція, яка дозволила будувати між кривими, з набагато більшим контролем за дотику (або до сполучних поверхонь, або до заданого

вектора). Знову ж таки, це здається невеликим оновленням, але якщо ви намагаєтеся відновити складну модель, то це буде знахідкою.

Що стосується пошкодження моделі, то також було виконано деяку роботу по прискоренню процесу видалення функцій, наприклад, для скорочення часу, витраченого на видалення великої кількості функцій малих отворів від хвилин до секунд. Як ми бачимо на рисунку 1.1, весь цей процес гарно візуалізований.

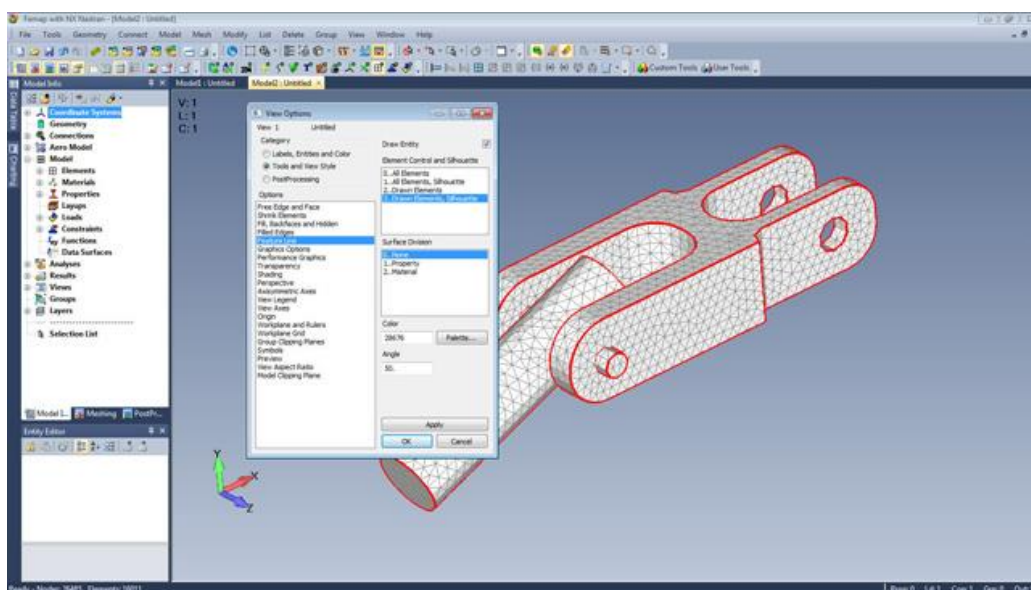


Рисунок 1.1 - Нова функція та силуетні лінії допомагають при візуалізації моделі, підкреслюючи краї та особливості моделі

Переходячи від обробки геометрії, ключовою частиною процесу попередньої обробки є створення сітки.

Як і очікувалося, Femap вже мав багатий набір інструментів для створення сіток для різноманітних типів досліджень і багато зробив для підвищення існуючої функціональності протягом багатьох років. Тим не менш, в останньому випуску ще є місце для нових інструментів і перероблених робочих процесів.

Одним з таких прикладів є нові засоби для миття і накладки; це допоможе вам створити відображені сітки навколо отворів і слотів, щоб поліпшити якість сітки, таким чином підвищуючи точність моделі. Хоча існували існуючі інструменти для роботи з найбільш поширеними круговими отворами в виробі, вони були розширені для роботи з некруглими отворами. Це ідеально підходить для нестандартних отворів, будь то для шайб, отворів у 3D-формах або предметів, таких як прокладки.

В іншому місці було цікаво працювати навколо повторного використання геометрії та мереж у більш ефективній формі. Тепер, при захопленні вибір геометрії, що має приєднану сітку, або можна перемістити, скопіювати або віддзеркалити її, він візьме і геометрію, і сітку.

Якщо є інші елементи, пов'язані з цим вибором (наприклад, навантаження, обмеження, регіони, що зв'язуються), вони також будуть повторювати їх. Така ж можливість була введена і для полярних масивів. Для тих користувачів, яким подобається стежити за тим, як об'єкти нумеруються, ці інструменти також будуть дотримуватися будь-яких діючих норм нумерації.

Що стосується схем нумерації, то також була зроблена робота щодо того, як обробляється нумерація об'єктів при побудові моделей з окремих файлів сітки.

Команда злиття моделей тепер надає повний контроль над тим, як обробляється нумерація під час процесу імпорту та відновлення, при з'єднанні розрізнені сіток в одну структуру.

Хоча ми розглядали більшість оновлень для попередньої обробки, в останніх випусках було також зроблено багато роботи з аспектів Femap для постобробки, як на рівні функціональності, так і в тому, як система підключається до інших системних рішень.

На фронті після обробки було зроблено роботу щодо відображення результатів. Наприклад, у той час як попередні версії використовували підхід до кольорових діапазонів "один розмір підходить для всіх", а їх застосування в наборі результатів тепер набагато більше.

Поряд з цим, побудова динамічних критеріїв дозволяє вибрати критерій і змінити його до вибраних меж або значень і відразу побачити, як виглядає модель на екрані.

Можна використовувати операцію "Створити з критерію" для групування прохідних елементів. Обидва способи значно полегшують перевірку складних моделей, дозволяючи отримати більш швидке розуміння результатів.

В іншому місці на фронті пост-обробки Femar отримує більш широкі інструменти для графіків в останніх кількох випусках, поза типовими інструментами X / Y.

Ідея полягає в тому, що замість використання іншої системи (наприклад, Microsoft Excel) можна працювати всередині Femar, підтримувати її зв'язок з вашими даними і продовжувати оновлювати в одному місці.

Нарешті, в пост-обробці функціональність, система нарешті отримала свій власний інструмент автоматичного формування звітів. Ми всі знаємо, що це означає: здатність налаштовувати ділянки та графіки, а система має витягувати досить загальний звіт.

Ці речі зазвичай не є занадто простими, але принаймні дають вам можливість швидко створити всі свої ділянки за один раз.

Femar відмовляється від типового підходу HTML і замість цього надає звіт у форматі Microsoft Word, зберігаючи ще один крок на цьому шляху.

Це ПЗ має багатий вибір для оптимізації продуктивності. На рисунку 1.2 видно оптимізації топології для швидшого розрахунку пружності кожної частини деталі, що моделюється.

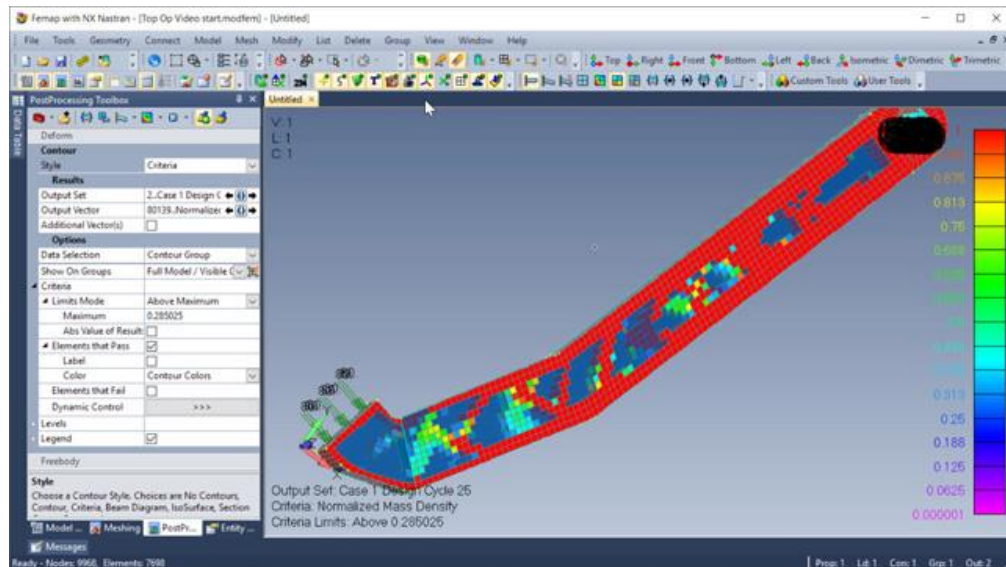


Рисунок 1.2 - Оптимізація топології розширює можливості рішення для оптимізації продуктивності проектування

1.3 Обґрунтування теми дипломного проекту

Тема дипломного проекту це розробка свого примітивного аналогу Femap, що базується на інших ЧМ. ПЗ цього типу зараз активно використовується в будь-якій промисловості, оскільки є дуже економічно вигідним. Майже всі великі компанії використовують моделювання замість експерименту, оскільки це значно дешевше і швидше, а також дозволяє відсіяти багато проблем на стадії розробки продукту.

Оскільки багато рішень не мають чіткої документації, а великі комерційні рішення часто бувають надлишковими, то даний проект має за мету зробити нескладний зрозумілий продукт для швидких підрахунків простих тіл.

На разі рішень які базуються на ЧМ багато, оскільки майже будь-який параметр системи можна вирахувати зараз и на майбутнє. Автомобілі, літаки, трубопроводи, будинки або будівельні споруди – все це можна зробити надійними завдяки даному ПЗ. Складність полягає лише в тому, що перевірити ці розрахунки людині дуже складно, оскільки ДР 4 порядку вже достатньо складні для того щоб людина могла їх розв’язати за деякий нормальний в рамках виробництва час, тому треба сподіватися на досвід розробника і тестування програми.

ПЗ, розроблене в рамках цього проекту може вираховувати нескладні фізичні тіла, які можна розписати системою 4 порядку. Основна його задача змоделювати, зрозуміти та протестувати сполучення методів Рунге-Кутта, стрільби та дискретної ортогоналізації. Воно не має графічної оболонки для більшої наглядності, оскільки розробка графічної оболонки, що буде сприймати коефіцієнти, є складною та окремою задачею. З точки зору методу скінченних елементів їх результат значно простіше використати для графічної візуалізації, оскільки це метод значно більш складний і професійний, проте він значно ресурсозатратніший.

На базі метода Рунге-Кутта створено багато методів, а також імплементовано багато рішень, оскільки саме цей метод вирішує одну з найпопулярніших задач що формуються на першому етапі – на моделюванні, це розрахунок системи звичайних диференціальних рівнянь (СЗДР). Інші методи використовуються для вирішення крайової задачі, що є задачею Коші, на кінцях інтервалів, що розглядаються. Крайова задача, це наслідок розвитку чисельних методів.

В цій роботі використовується метод Рунге-Кутта 4 порядку, часто коли говорять про загальний метод Рунге-Кутта, мають на увазі саме його. Також важливою частиною є дискретна ортогоналізація, яка є достатньо незвичайним прийомом, і досі не має теоретичного обґрунтування, проте гарно показує себе з практичної точки зору. Цей метод відноситься до методів

що використовують прийом ортогоналізації для отримання та уточнення результату. Даний метод тільки починають використовувати в FEA-пакетах.

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		20

2. ТЕХНОЛОГІЧНІ ТА ПРОГРАМНІ ЗАСОБИ РОЗРОБКИ

2.1 Класифікація ЧМ

Існує велика кількість численних методів, які можливо ділити за принципом дії, швидкістю роботи, точністю результату, оптимальним алгоритмом або складністю реалізації. За допомогою чисельних методів вирішують різні задачі:

- Розв'язок лінійних та нелінійних рівнянь та їх систем;
- Інтерполяція та апроксимація функцій;
- Числове інтегрування та обчислення похідної;
- Чисельний розв'язок диференціальних рівнянь (ДР) та їх систем;
- Чисельний розв'язок систем інтегральних рівнянь;
- Задачі оптимізації.

Найчастіше ЧМ ґрунтуються на інтерполюванні, ортогоналізації, врівноваженні рівнянь, умовній мінімізації тощо. Серед них на даний момент найбільш широко вживаються методи кусково-многочленної та раціональної сплайнової апроксимації, а також адаптивної апроксимації та нелінійної за параметром апроксимації.

ЧМ потрібні для того щоб замінити експеримент, тобто замість створення дослідної установки, запуску її та заміру результатів які потрібні в рамках дослідження, чисельні методи дозволяють зробити математичну модель, що дозволяє вирахувати потрібні показники не вдаючись до реальних розробок. Спочатку робиться постановка задачі, як правило це формулювання системи рівнянь, що описують процес, з коефіцієнтами, що є характеристиками для цього процесу[12]. Звичайно, математичка модель не може бути всебічною, тому вона проектується для знаходження конкретних величин згідно з постановкою задачі. Правильність використання чисельних методів сильно спирається на правильність математичної моделі, без правильної моделі вони просто не мають жодної цінності. На другому етапі рішення проблеми

використовується так зване математичне дослідження [12]. Це і є використання чисельних методів для вирішення задачі сформульованої при першому етапі. Також на другому етапі використовують аналітичні рішення, але вони часто бувають неточними, а також не всюди є і навіть можливими. Більш точними є математичні методи, але вони також не завжди відповідають запитам, оскільки не завжди можуть розрахувати складні задачі. Проте, чисельні методи мають більш складну реалізацію, тому до них не завжди звертаються. При моделюванні вимоги до самої математичної моделі достатньо високі – вона має бути в 2-4 рази точніша ніж реальна фізична модель, для того щоб результати були точними, і могли використовуватись. Більше ніж в 2-4 рази збільшувати точність сенсу немає, оскільки на результат і його точність це вже не впливає. Знижати точність не можна, оскільки за рахунок вирішення багатьох додаткових систем, рівнянь тощо, може статись дуже велика похибка.

Однією з великих проблем постає перевірка алгоритму чисельного методу та його реалізації [12]. Оскільки чисельні методи на відміну від класичної математики базуються на інших засадах, і тому чітко перевірити наскільки воно якісно працює, и чи працює взагалі. За рахунок складності чисельних методів, при помилці в моделі похибка перетворюється на помилковий результат.

Результат розрахунку чисельного метода завжди перевіряють на даних реального експерименту, що проводять після отримання бажаних результатів шляхом моделювання.

На даний момент ЧМ, за рахунок гарного розвитку електронно-обчислювальних машин, не мають проблем за апаратною складовою, єдина їх проблема це правильна розробка та використання.

Розглянемо чисельні методи, на яких базуються комплекси ПЗ для задач аналізу твердого деформованого тіла.

2.2 Алгоритм методу Рунге-Кутта

Метод Рунге-Кутта – це великий підрозділ ЧМ для вирішення задачі Коші для звичайних диференційованих рівнянь або системи звичайних диференційованих рівнянь.

Найбільш вживаним є метод Рунге-Кутта 4-ого порядку [1]. Формулюється задача Коші для системи звичайних диференціальних рівнянь (СЗДР) довільного порядку в векторній формі.

$$x' = f(x, y)$$

$$y(x_0) = y_0$$

Далі використовується ітеративна формула для розрахунку членів ряду,

$$y_{i+1} = y_m + \frac{h}{6} (k_1 + 2k_2 + 2k_3 + k_4)$$

де кожний коефіцієнт вираховується за формулою:

$$k_1 = f(x_i, y_i),$$

$$k_2 = f\left(x_i + \frac{h}{2}, y_i + \frac{hk_1}{2}\right),$$

$$k_3 = f\left(x_i + \frac{h}{2}, y_i + \frac{hk_3}{2}\right),$$

$$k_4 = f(x_i + h, y_i + hk_3).$$

Цей метод є настільки вживаним, що часто не уточнюють що використовується саме метод Рунге-Кутта 4 порядку і говорять просто метод Рунге-Кутта.

2.3 Алгоритм методу стрільби

Алгоритм методу стрільби був придуманий для вирішення крайової задачі, шляхом зведення її до задачі Коші [2, 11].

Базовий принцип будується на тому що будь-яка крайова задача для системи диференціальних рівнянь може бути замінена задачею Коші.

$$\begin{cases} y_1' = f_1(x, y_1, y_2, \dots, y_n), \\ \dots \\ y_n' = f_n(x, y_1, y_2, \dots, y_n). \end{cases}$$

Для того щоб досягнути такої заміни рішення задачі Коші трохи модифікується.

$$\begin{cases} y_1 = f_1(x, c_1, c_2, \dots, c_n), \\ \dots \\ y_n = f_n(x, c_1, \dots, c_n), \end{cases}$$

А сама функція виводиться як функція залежна від введених постійних коефіцієнтів.

$$\begin{cases} y_1(0) = c_1 \\ \dots \\ y_n(0) = c_n \end{cases}$$

Таким чином, формується система алгебраїчних рівнянь відносно постійних коефіцієнтів.

$$\begin{cases} \varphi_1(y_1(l, c_1, \dots, c_n), y_n(l, c_1, \dots, c_n)) = 0 \\ \dots \\ \varphi_k(y_1(l, c_1, \dots, c_n), y_n(l, c_1, \dots, c_n)) = 0 \end{cases}$$

Якщо записати крайову задачу в матричній формі, вона буде виглядати так:

$$y' = A(x)y + f(x), (0 \leq x \leq l)$$

де,

$$y = \begin{pmatrix} y_1 \\ y_2 \\ \dots \\ y_n \end{pmatrix}, A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix}$$

При:

$$P = \begin{pmatrix} p_{11} & p_{12} & \dots & p_{1n} \\ p_{s1} & p_{s2} & \dots & p_{sn} \end{pmatrix}, \theta = \begin{pmatrix} \theta_1 \\ \theta \end{pmatrix}, Q = \begin{pmatrix} q_{11} & q_{12} & \dots & q_{1n} \\ p_{s1} & p_{s2} & \dots & p_{sn} \end{pmatrix}, \gamma = \begin{pmatrix} \gamma_1 \\ \gamma_k \end{pmatrix}$$

Еквівалентна задача Коші буде виглядати для неї так:

$$y' = A(x)y + f(x), (0 \leq x \leq l)$$

де,

$$c = (c_1, c_2, \dots, c_n)^T$$

Підстановка вектора постійних коефіцієнтів дає дві системи алгебраїчних рівнянь (САР) , які вже легко вирішується.

$$\begin{cases} p_{11}c_1 + p_{12}c_2 + p_{1n}c_n = \theta_1, \\ p_{s1}c_1 + p_{s2}c_2 + p_{sn}c_n = \theta_s \\ \begin{cases} q_{11}y_1(l, c_1, \dots, c_n) + \dots + q_{1n}y_n(l, c_1, \dots, c_n) = \gamma_1, \\ q_{k1}y_1(l, c_1, \dots, c_n) + \dots + q_{kn}y_n(l, c_1, \dots, c_n) = \gamma_k \end{cases} \end{cases}$$

2.4 Алгоритм методу скінченних елементів [1,2]

На зараз це основний метод сучасної будівельної механіки, що лежить в основі переважної більшості сучасних програмних комплексів, призначених для виконання розрахунків будівельних конструкцій на ЕОМ.

Але діапазон його застосування надзвичайно широкий: будівництво і машинобудування, гідро- і аеродинаміка, гірнича справа і новітня техніка, а також різні завдання математичної фізики - теплопровідності, фільтрації, поширення хвиль тощо.

Метод скінченних елементів (МСЕ) вперше був застосований в інженерній практиці на початку 50-х рр. XX ст. Спочатку він розвивався а двома незалежними один від іншого напрямками - інженерним і математичним. На ранньому етапі формулювання МСЕ ґрунтувалися на принципах будівельної механіки, що обмежувало сферу його застосування. І тільки коли були сформульовані основи методу в варіаційної формі, стало можливим поширення його на багато інших завдань. Швидкий розвиток МСЕ йшло паралельно з прогресом сучасної комп'ютерної техніки і її застосуванням у різних областях науки та інженерної практики.

Значний внесок у розробку МСЕ був зроблений Дж. Аргірісом [11]. Їм вперше дано загальне матричне формулювання розрахунку стержневих

систем на базі фундаментальних енергетичних принципів, визначена матриця податливості, а також введено поняття матриці жорсткості (як зворотної матриці податливості). Роботи Дж. Аргіріса і його співробітників, опубліковані в період 1954-1960 рр., дали відправну точку для матричного формулювання відомих чисельних методів і застосування ЕОМ в розрахунках конструкцій.

Для розвитку МСЕ особливе значення мали варіаційні принципи механіки і математичні методи, засновані на цих принципах. Дискретизацію завдання на основі варіаційного методу Рітца вперше в 1943 р застосував Р. Курант. Лише в 50-і рр. з'явилися аналогічні роботи Ж. Полі, Ж. Герша і ін.

Перша робота, в якій була викладена сучасна концепція МСЕ, відноситься до 1956 р Американські вчені М. Тернер, Р. Клафф, Г. Мартін і Л. Топп, вирішуючи плоску задачу теорії пружності, ввели елемент трикутного виду, для якого сформували матрицю жорсткості і вектор вузлових сил. Назва - метод кінцевих елементів ввів в 1960 р Р. Клафф. В період 1960-1965 рр. опубліковані роботи, в яких на основі варіаційних принципів отримані кінцеві елементи для вирішення завдань вигину плит, тонких оболонок, масивів. Серед них можна відзначити роботи Р. Мак-Лейа, Р. Мелош, Дж. Бесселіна, Ф. де Веубеке, М. Джонса, Т. Пиана. У 1967 р видана перша монографія про МСЕ О. Зенкевича і І. Чанга, в якій викладені основи методу і області його застосування.

До сімдесятих років відноситься поява математичної теорії кінцевих елементів. Тут можна виділити праці І. Бабусі, Р. Галлагера, Ж. Дек-лу, Дж. Одена, Г. Стренга, Дж. Фікса. Значний внесок у розробку теоретичних основ МСЕ внесли і російські вчені. В. Г. Корнеєв вказав на збіг математичної суті МСЕ і ВРМ. Зіставлення МСЕ з рядом варіаційних методів наведено в працях Л. А. Розіна. Під керівництвом А. С. Сахарова розроблена моментна схема скінченних елементів.

Період останніх десятиліть особливо характерний для розвитку і застосування МСЕ в таких областях механіки суцільних середовищ, як оптимальне проектування, облік нелінійного поведіння, динаміка конструкцій тощо.

Метод кінцевих елементів, як і багато інших чисельні методи, заснований на поданні реальної континуальної конструкції її дискретної моделлю і заміні диференціальних рівнянь, що описують ПДВ суцільних тіл, системою алгебраїчних рівнянь. Разом з тим МСЕ допускає ясну геометричну, конструктивну і фізичну інтерпретацію.

Суть методу полягає в тому, що область (одно-, дво- або тривимірна), яку займає конструкцією, розбивається на деяке число малих, але скінченних за розмірами підобластей (рисунки 1.3). Останні носять назву скінченних елементів (КЕ), а сам процес розбиття - дискретизацією.

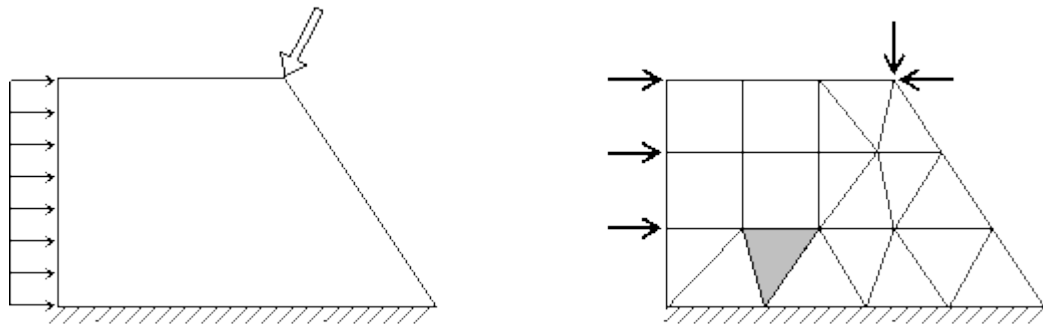


Рисунок 1.3 – Розбиття фігури на скінченні елементи

Залежно від типу конструкції і характеру її деформації КЕ можуть мати різну форму. Так, при розрахунку стрижневих систем (ферми, балки, рами) КЕ є ділянками стержнів; для двовимірних континуальних конструкцій (пластини, плити, оболонки) найчастіше застосовують трикутні і прямокутні (плоскі або вигнуті) КЕ; а для тривимірних областей (товсті плити, масиви) - КЕ у формі тетраедра або паралелепіпеда. На відміну від реального споруди в дискретної моделі кінцеві елементи зв'язуються між собою тільки в окремих точках (вузлах) визначеним кінцевим числом вузлових параметрів.

МСЕ - це варіаційний метод. Функціонал енергії для всієї розглянутої області тут представляється у вигляді суми функціоналів окремих її частин - кінцевих елементів. По області кожного елемента, незалежно від інших, задається свій закон розподілу шуканих функцій. Така кусочно-безперервна апроксимація виконується за допомогою спеціально підібраних апроксимуючих функцій, які називаються також координатними або таким, що інтерполуються. З їх допомогою шукані безперервні величини (переміщення, напруги тощо) в межах кожного КЕ виражаються через значення цих величин в вузлових точках, а довільна задана навантаження замінюється системою еквівалентних вузлових сил.

При такій кусочно-безперервною апроксимації забезпечується умова спільності лише в вузлах, а в інших точках на межі КЕ ця умова задовольняється в загальному випадку наближено (в зв'язку з цим розрізняють КЕ різного ступеня спільності).

Найбільшого поширення набув метод кінцевих елементів в переміщеннях, що має багато спільного з методом Рітца і варіаційно-різницеvim методом (надалі ми будемо в основному розглядати саме цей варіант МСЕ) [1]. Різниця між традиційною схемою методу Рітца і МСЕ в формі методу переміщень полягає у виборі системи апроксимуючих функцій. Якщо в методі Рітца апроксимація переміщень проводиться по всій області їх визначення, то в МСЕ - по кожному кінцевому елементу окремо, що дозволяє використовувати апроксимуючі функції більш простого виду. У першому випадку функціонал повної потенційної енергії варіюється по невизначеним коефіцієнтам, у другому - по переміщенням в вузлах сітки, що призводить до системи алгебраїчних рівнянь методу переміщень (основними невідомими є безпосередньо вузлові переміщення). При цьому використання кусочно-безперервною апроксимації дозволяє отримати рідко заповнену або стрічкову

структуру матриці коефіцієнтів системи рівнянь і таким чином дає можливість застосування більш ефективних методів її рішення.

Число вузлів і число переміщень в вузлі (ступінь свободи вузла), прийняті для кінцевого елемента, можуть бути різними, однак не повинні бути менше мінімально необхідних для опису напружено-деформованого стану КЕ в рамках прийнятої фізичної моделі. Число незалежних переміщень у всіх вузлах елемента визначає ступінь свободи КЕ. Ступінь свободи всієї конструкції і відповідно порядок системи дозволяють рівнянь визначається сумарним числом переміщень всіх її вузлів. Оскільки основними невідомими МСЕ в формі методу переміщень вважаються вузлові переміщення, ступінь свободи КЕ і всієї конструкції в цілому є надзвичайно важливим поняттям в МСЕ. Поняття про ступінь свободи вузла, КЕ і конструкції і ступеня їх же кінематичної невизначеності ідентичні.

Спосіб розбивки даній області на кінцеві елементи, їх число і число ступенів свободи, а також вид апроксимуючих функцій в кінцевому підсумку визначають точність розрахунку конструкції. Слід зазначити, що простим збільшенням числа кінцевих елементів не завжди вдається досягти підвищення точності розрахунків. Питання стійкості і збіжності рішення, а також оцінки точності отриманих результатів є основними при використанні МСЕ.

У порівнянні з іншими чисельними методами МСЕ в кращому ступені алгоритмізувати і більш гнучкий при описі геометрії і граничних умов розраховується області. Крім того, до переваг методу слід віднести його фізичну наочність і універсальність.

Стосовно до стрижневих систем МСЕ в формі методу переміщень може розглядатися як матрична форма класичного методу переміщень, що

відрізняється тільки більш глибокої формалізацією алгоритму і орієнтацією його на використання ЕОМ. [2]

2.5 Приклад реалізації методу Рунге-Кутта на мові Fortran

```
1  program RK4
2  use NML
3  implicit none
4  integer, parameter:: m=2
5  real:: A, B, Y(m), DY(m), E(m)
6  integer:: i, n, cnt=0
7  !begin
8      A=0.0; B=4.0; n=256
9      Y(1)=1.0; DY(1)=0.0
10     Y(2)=0.0; DY(2)=0.5
11     E(1)=B+exp(-B)
12     E(2)=0.5*B*exp(2.0*B)
13
14     call DE30(DS2, A, B, n, Y, DY)
15     print 10, Y, E, cnt
16 10  format(/'      Y1 =',F10.6,'      Y2 =',F14.6      &
17         /'      E1 =',F10.6,'      E2 =',F14.6      &
18         /'      cnt =',I5)
19
20  contains
21  subroutine DS2(X, Y, DY, DDY)
22  real, intent(in):: X, Y(:), DY(:)
23  real:: DDY(:)
24  !begin
25      DDY(1)=DY(1)+2.0*Y(1)-4.0*Y(2)*exp(-2.0*X)-1.0
26      DDY(2)=2.0*DY(2)+(Y(1)-X)*exp(3.0*X)
27      cnt=cnt+1
28      return
29  end subroutine DS2
```

Рисунок 2.1 – Реалізація методу Рунге-Кутта мовою Fortran

Написаний метод Рунге-Кутта на мові Fortran працює швидше за всіх (~0.22 секунди). Реалізація виглядає трохи складною, і як і усі мови програмування такого типу Fortran-код виконується дуже швидко, досить проста і зрозуміла реалізація з легким відлагодженням. Проте, неможливо його майже візуалізувати, а також майже у всі сучасні системи неможливо додати такі проекти.

2.6 Приклад реалізації методу Рунге-Кутта на мові C/C++

```
#include <iostream>
#include <cmath>
using namespace std;

double F(double x, double y){
    return 3*sin(2*y)+x;
}

int main() {
    double a=0; double b=1; double h=0.1;
    double n=(b-a)/h;
    double X[(int)n];
    double Y1[(int)n];
    double Y2[(int)n];
    double Y3[(int)n];
    double Y4[(int)n];
    double Y[(int)n];
    //calculate
    X[0]=a; Y[0]=2;
    for(int i=1; i<=n; i++){
        X[i]=a+i*h;
        Y1[i]=h*F(X[i-1],Y[i-1]);
        Y2[i]=h*F(X[i-1]+h/2.0,Y[i-1]+Y1[i]/2.0);
        Y3[i]=h*F(X[i-1]+h/2,Y[i-1]+Y2[i]/2);
        Y4[i]=h*F(X[i-1]+h,Y[i-1]+Y3[i]);
        Y[i]=Y[i-1]+(Y1[i]+2*Y2[i]+2*Y3[i]+Y4[i])/6;
    }
    //print results
    for(int i=0; i<=n; i++){
        cout << "X["<<i<<"]="<<X[i] << " ";
    }
    cout << endl;
    for(int i=0; i<=n; i++){
        cout << "Y["<<i<<"]="<<Y[i] << " ";
    }
    return 0;
}
```

Рисунок 2.2 – Реалізація методу Рунге-Кутта мовою C++

Більш наочна і очевидна реалізація ніж в Fortran, також можливо вставити як модуль майже в будь-яку систему або візуалізувати. Проте працює за 0.3 секунди в середньому, що гірше ніж попередній варіант.

2.7 Приклад реалізації методу Рунге-Кутта на мові Python

```

1  def RK4(f):
2      return lambda t, y, dt: (
3          lambda dy1: (
4              lambda dy2: (
5                  lambda dy3: (
6                      lambda dy4: (dy1 + 2*dy2 + 2*dy3 + dy4)/6
7                          )( dt * f( t + dt, y + dy3 ) )
8                      )( dt * f( t + dt/2, y + dy2/2 ) )
9                      )( dt * f( t + dt/2, y + dy1/2 ) )
10                     )( dt * f( t, y ) )
11
12  def theory(t): return (t**2 + 4)**2 /16
13
14  from math import sqrt
15  dy = RK4(lambda t, y: t*sqrt(y))
16
17  t, y, dt = 0., 1., .1
18  while t <= 10:
19      if abs(round(t) - t) < 1e-5:
20          print("y(%2.1f)\t= %4.6f \t error: %4.6g" % ( t, y, abs(y - theory(t))))
21          t, y = t + dt, y + dy( t, y, dt )
22

```

Рисунок 2.3 – Реалізація методу Рунге-Кутта мовою Python

На мові програмування Python найпростіша реалізація та інтеграція в систему, а також набагато легша візуалізація, проте за рахунок особливості мови працює вона в середньому ~ 0.7-0.8 секунд.

2.8 Порівняльна характеристика мов програмування для даної задачі

Як можна побачити з лістингу 1.1, на мові Fortran за рахунок його низькорівневої природи, виходить достатньо об'ємна реалізація, проте за рахунок того що він майже «напрямую» трансліюється в інструкції, найбільш висока швидкодія. На Visual Fortran достатньо небагатий вибір для створення графічного інтерфейсу, тому найбільш прагматичним рішенням є виводити дані в файл.

На мові C++ реалізація стає набагато більш зрозумілою, але за рахунок передачі великих даних з функції в функцію і складності обчислень програма працює повільніше. Однак інтегрування в деякий продукт значно легше, а також можливо вже зробити користувацький інтерфейс, однак потрібно добавляти достатньо багато додаткових ресурсів, що теж не гарно впливає на швидкодію.

На мові Python реалізація є найбільш наочною. Проте, за рахунок особливостей скриптованих мов швидкодія є найповільнішою з усіх. Для інтегрування в деякі системи Python є найпростішим варіантом, в деякі це неможливо. Багатий вибір можливостей для інтерфейсу користувача, у тому числі і дуже наразі популярний веб-інтерфейс.

Оскільки в даному проекті нас цікавить максимальна швидкодія і простота реалізації, вибір за Fortran, хоча на C++ реалізація простіша, проте відлагодження програми, яке є дуже важливим, значно складніше, що і послугувало причиною вибору в напрямку Fortran. Відлагодження програми яка використовує чисельні методи є непростю задачею, і тому чим простіше це зробити, тим вигіднішим є обрана технологія. Як ми бачили в главі 1, достатньо багато програм використовують саме Fortran-ядро для своїх програм, оскільки для комерційного продукту швидкодія і можливість «безболісного» відлагодження є необхідністю.

Багато сучасних FEA-пакетів, не дивлячись на популярність мов програмування Python/JS, навіть зараз мають основну логіку продукту прописану на старій мові програмування Fortran, оскільки деякі проекти не хочуть змінювати стек технологій, а деякі потребують максимальної швидкодії. Звичайно, з точки зору розуміння коду, Fortran програє будь-якій мові, оскільки в реалізації Fortran будівництво архітектури реалізоване на мітках, а навіть в такій невеликій програмі, більше 2000 міток, на які постійно є свої переходи. Це значно ускладнює задачу, тому що таку кількість

переходів майже неможливо оцінити та повністю з'ясувати загальну будову програми.

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		35

3. ПРОГРАМА ДЛЯ РОЗРАХУНКУ ЗАДАЧІ ТВЕРДОГО ДЕФОРМОВАНОГО ТІЛА

3.1 Загальна структура програми

Ця програма є збіркою реалізацією декількох численних методів що обмінюються даними між собою. Базові значення, тобто вхідні дані які використовує програма для розрахунків, а також місце куди буде збережений результат об'являються наступним чином:

```
integer, parameter::n=6,k=3,nl=3,np=3
dimension z(N*(N-K+1)+(N-K+1)*(N-K+2)/2+1),
*r1(N*(N-K+1)),z1(N*(N-K+1)),y(n+1),a1(N*(N+1)),
*om((N-K+1)*(N-K+2)/2),
xaindl(nl), aindp(np),ell(nl),elp(np),itp(n)
real
Aleft(n*(n+1)),Aright(N*(N+1)),AMat1(n/2,n+1),Amat2(n/2,n+1)
open (6,file='dior.txt',status='unknown')
x=0.
```

Лістинг 1.1 – Вхідні дані програми

де,

- N – порядок системи рівнянь;
- K – число лівих граничних умов;
- NL – число ненульових коефіцієнтів рівнянь лівих граничних рівнянь;
- NP - число ненульових коефіцієнтів рівнянь правих граничних рівнянь;
- X – початкове значення аргумента;
- AINDL – масив з значеннями індексів ненульових елементів лівих граничних умов;
- AINDP - масив з значеннями індексів ненульових елементів правих граничних умов;

- ELL – масив елементів значень ненульових коефіцієнтів розширеної матриці лівих граничних умов;
- ELP – масив елементів значень ненульових коефіцієнтів розширеної матриці правих граничних умов.

Програма складається з функції Do, що є основною функцією програми, в ній закладений цикл та послідовність виконання дій, та підфункцій System, Bacond, Forwar, Matrix та Result. В функції Do вони послідовно визиваються, і використовують дані попередньої підфункції. Програма записує результати в файл.

```
open (6,file='dior.txt',status='unknown')
```

Лістинг 1.2 – Збереження даних у файл

Всі вхідні дані конфігуруються через значення початкових змінних в коді. По коефіцієнтам які дає ця програма можливо далі змодельовати деформацію тіла, проте для візуалізації цього потрібно реалізувати складний рендерінг об'ємної фігури, а для задачі поставленої для даного проекту це є надлишковим. Самі коефіцієнти подаються у вигляді чисел в форматі чисел з плаваючою комою.

```
0.10000E+02 -0.36380E-10 0.25692E+01 -0.27929E+00 0.00000E+00
0.61846E-10 -0.71800E+00
```

```
0.99000E+01 0.13985E+02 0.25692E+01 -0.21209E+00 -0.23237E+01
0.26746E+03 -0.62618E+00
```

```
0.98000E+01 0.51213E+02 0.25692E+01 -0.15396E+00 -0.45689E+01
0.46651E+03 -0.53698E+00
```

```
0.97000E+01 0.10533E+03 0.25692E+01 -0.10454E+00 -0.66756E+01
0.60689E+03 -0.45240E+00
```

0.96000E+01 0.17094E+03 0.25692E+01 -0.63282E-01 -0.86014E+01
 0.69780E+03 -0.37384E+00
 0.95000E+01 0.24353E+03 0.25692E+01 -0.29538E-01 -0.10319E+02
 0.74783E+03 -0.30223E+00
 0.94000E+01 0.31941E+03 0.25692E+01 -0.25861E-02 -0.11813E+02
 0.76479E+03 -0.23807E+00
 0.93000E+01 0.39562E+03 0.25692E+01 0.18331E-01 -0.13078E+02
 0.75568E+03 -0.18154E+00
 0.92000E+01 0.46988E+03 0.25692E+01 0.33974E-01 -0.14117E+02
 0.72667E+03 -0.13255E+00

Лістинг 1.3 – Приклад виводу програми

Кожен коефіцієнт відповідає за зміну деякого «вузла», і по цим даним змінюється його місцеположення, пружність при наступній деформації тощо. Тіло яке розглядається для прикладу в даній програмі – це усічений циліндр.

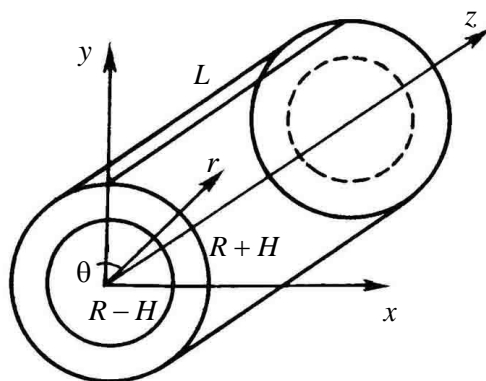


Рис. 3.1 – Тіло для задачі

Функції в програмі реалізовані як сабрутини, що дозволяє їм вертатися на ту мітку з якої вони були викликані.

```

subroutine result(n,ni,a)

dimension a(n)

write(6,1)a

1    format (7(e12.5,2x))

return

end

```

Лістинг 1.4 – Приклад малої сабрутини

Програма не використовує додаткових бібліотек, лише базові операції що представлені мовою Fortran. Також програма використовує базові конструкції та засоби встановлення порядку виконання команд, такі як виклики підфункцій,

```
call bacond(n,k,i1,ii,nl,aindl,itp,ell,a1)
```

Лістинг 1.5 – Приклад виклику підпроцедури

циклів з лічильником,

```

do i=1,n/2

amat1(i,i)=1.

Enddo

```

Лістинг 1.6 - Приклад циклу з лічильником

а також стрибки на мітки.

```
if (i1.ne.0) goto 8
```

Лістинг 1.7 – Стрибок на мітку

```
6 amax=a(ji)
```

Лістинг 1.11 – Об’ява мітки

Структури даних, що використовуються – це масиви,

dimension z(N*(N-K+1)+(N-K+1)*(N-K+2)/2+1)

Лістинг 1.12 – Об’явлення масиву

цілі числа,

integer, parameter::n=6,k=3,nl=3,np=3

Лістинг 1.13 – Об’явлення цілих змінних

а також дійсні числа.

real Aleft

Лістинг 1.14 – Об’явлення дійсної змінної

3.2 Функціонування компонент програми

Першою компонентою цього ПЗ є функція system, яка вирішує систему лінійних алгебраїчних рівнянь (СЛАР) одночасно с декількома правими частинами методом Гауса з вибором головного елемента. Складність такого алгоритму є $O(n^3)$ для порядку системи рівнянь $n > 50$, тому його недоцільно використовувати для систем високих рангів. Оскільки порядок нашої системи як правило $n \approx 4$, то складність алгоритму падає до $O(n^2)$, що, враховуючи велику точність та гнучкість цього алгоритму, дає нам право стверджувати що він є оптимальним. На відміну від інших методів, він залежить лише від порядку системи рівнянь, в той час як інші методи ще й залежать від насиченості матриці, кількості ненульових елементів тощо.

Функція `forwar` відповідає за управління прямим ходом вирішення крайової задачі. Крайова задача вирішується в два етапи, при прямому ході та оберненому. Ця функція відповідає за прогонку крайової задачі при прямому ході. На прямому ході вирішується задача Коші для визначаючих вектор-функцій і знаходженні фундаментальних матриць рішень на кожному з відрізків. Зворотній же хід є рішенням СЛАР, яку ми отримуємо внаслідок прямого проходу крайової задачі.

Функція `matrix` відповідає за розрахунок елементів матриці коефіцієнтів. Вона викликається для розрахунку коефіцієнтів фундаментальної матриці яку ми отримуємо внаслідок прямого ходу крайової задачі.

Функція `basond` в свою чергу відповідає за вирахування коефіцієнтів систем рівнянь лівих та правих граничних умов. Тобто, фактично вона є підготовкою до зворотного ходу крайової задачі.

Основну частину програми складає функція `do`, що за допомогою метода дискретної ортогоналізації вирішує крайову задачу. Метод дискретної ортогоналізації був створений для рішення двухточкових крайових задач для СЗДР. Він базується на ортогоналізації компонент рішення в дискретному наборі точок інтервалу цього рішення. На даний момент цей метод не має фізичного чіткого пояснення, і тому не має чіткого теоретичного аналізу. Проте на практиці цей метод заявив себе як один з найбільш перспективних на сьогоднішній день. По факту він складається з методу Рунге-Кутта, а також невеликої частини ортогоналізації Грама-Шмідта. Для його промислової реалізації існує пакет GMDO [11]. В випадку цього ПЗ, цей метод реалізований саме на базі методу Рунге-Кутта 4 порядку, для забезпечення задовільної точності і простоти обчислень.

Схему послідовних викликів можна побачити в додатку 1, структурній схемі роботи програми. Компоненти взаємодіють між собою через передачу

параметрів в підпрограми, вхідні дані є глобальними в області вихідного коду. Загалом програмне забезпечення, яке представлене в даній роботі показує оптимальну точність і швидкодію для складності тіла, що аналізується (рисунок 3.1).

3.3 Аналіз результатів

Результатом є таблиця коефіцієнтів, що показує зміну координат деяких вузлів тіла, а також його фізичних властивостей, таких як пружність статична, динамічна тощо. Як виглядають результати видно на лістингу 1.3, від складності тіла залежить кількість рядків файлу виводу, для цієї конкретної задачі – це 101 рядок. Точність коефіцієнтів досягає 7 знаків після коми, що є достатньо високим показником для цієї групи методів, що використовуються у цій програмі. На відміну від популярного метода скінченних елементів, що розбиває тіло на задану кількість фрагментів, фрагментація твердого тіла в даному ПЗ відбувається автоматично в залежності від складності його описання. Це дозволяє не думати про підбір оптимальної кількості елементів для розбивання. Програма, представлена у цій роботі працює досить швидко, і тому складно заміряти її швидкодію, можна сказати що час, за який вона виконується складає близько 0.1 секунди. Для порівняння, всього лише один метод Рунге-Кутта, імплементований на мові Python, займає близько 0.7 секунд.

ПЗ складається з методів, які підібрані оптимально саме під задачу такої складності, і не буде показувати гарні результати на тілах, для описання яких потребуються системи рівнянь порядку більше 50. Більш того, структури даних, якими оперує ця програма не розраховані на складні композитні тіла, такі як кузов автомобіля або складна механічна деталь, оскільки ця програма створена з метою ознайомлення з даним класом чисельних методів і задач які вони вирішують. Проте для неї можливо написати графічну оболонку на будь-якій зручній мові, котра буде використовувати результати в файлі результатів.

Задаючи різні вектори навантаження на тверде тіло, можна отримати деякі частини його незмінним або майже незмінними. В цьому конкретному прикладі ми маємо в останньому рядку два 0, що говорять про незмінні фізичні властивості пружності внаслідок деформації тіла, а також майже незмінене положення у просторі.

0.00000E+00 0.00000E+00 0.25692E+01 -0.27929E+00 0.00000E+00
0.00000E+00 0.71800E+00

Лістинг 1.15 – Приклад частково зміненого вузла тіла

ВИСНОВКИ

Задачею цього дипломного проекту було створення своєї примітивної програми, що вирішує задачі FEA-пакету. Також в цій роботі було проаналізовано ринок комплексів ПЗ для вирішення задач аналізу деформації твердого тіла, і методи на яких ці комплекси базуються.

По результатам проведеного аналізу було з'ясовано, що ринок на сьогоднішній день має велике різноманіття різних програм для абсолютно різних інженерних задач. Також було з'ясовано, що всі FEA-пакети не мають між собою принципової різниці, оскільки базуються на одних й тих же чисельних методах. Основні відмінності полягають в стеку технологій, підтримці операційних систем та платформ, документації та практичних прикладах, зручності та гнучкості інтерфейсу, ціновій політиці тощо. Були проаналізовані комерційні рішення, і з'ясувалось що вони значно відрізняються якістю служби підтримки, документацією, доступністю на різних цільових платформах та актуальністю інтерфейсу тощо. Незалежні бенчмарки [7] показують що різниця між усіма сучасними комерційними пакетами не складає більш ніж 10% на показник. Було також розглянуто можливі причини таких відмінностей і принципова ідея чому вона не може бути більшою. Також були розглянуті проблеми розвитку продуктів вітчизняного походження, внаслідок цього аналізу було видно, що продукти, які були розроблені українською школою математиків, мають часто кращі характеристики, ніж всесвітньо відомі аналоги типу ANSYS.

Крім комерційних, були розглянуті некомерційні FEA-пакети [9], і з'ясувалось, що вони не гірші в багатьох випадках за комерційні, а в технічних характеристиках бувають і кращими за них. Проте великим недоліком для їх користування є наявність неповної документації, відсутність служби підтримки, відсутність гарантії якісної працездатності. Ці фактори не дають використовувати повсюди ці комплекси ПЗ для промисловості, проте є

					ІАЛЦ.045490.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підп.	Дата		44

виробники які роблять це на свій страх і ризик. Цікаво, що фактор гарантії якісної працездатності є як і недоліком так і перевагою цього типу ПЗ, оскільки відкритий вихідний код дає можливість побачити як працює FEA-пакет.

Серед всіх комплексів ПЗ окремо був виділений Femap, який зараз є стандартом цього класу пакетів. Зручний інтерфейс, недорога ліцензія, гнучкі можливості налаштувань та великий функціонал дає йому право мати такий статус. В київському політехнічному інституті саме ця програма використовується для вивчення даної галузі в механіко-машинобудівному інституті. На базі нього були розглянуті основні задачі, які повинно виконувати дане ПЗ, стандартний інтерфейс (так звана FEMAP оболонка використовується більш ніж на 60 FEA-пакетах), можливості і налаштування.

Проведений аналіз методів, якими користуються усі FEA-пакети. Майже будь-який комплекс ПЗ в даній галузі використовує метод Рунге-Кутта, який є широковживаним для даного класу задач. В програмі, яка була розроблена в рамках цього проекту, використовується метод Рунге-Кутта 4 порядку. Також в ній використовується метод стрільби та метод дискретної ортогоналізації. Була проведена порівняльна характеристика на прикладі методу Рунге-Кутта між стеками технологій на яких найчастіше розробляють продукти цього класу.

Програмне забезпечення, яке було спроектовано в цій роботі – це примітивний FEA-пакет, що вирішує задачу аналізу твердого деформованого тіла, в даному конкретному випадку це усічений циліндр. Це ПЗ слугує для базового розуміння реалізації комплексу чисельних методів, взаємодії між ними, підбору чисельних методів для коректної взаємодії між собою. Також в цій програмі можна побачити якими даними оперують графічні програми, що візуалізують деформацію твердого тіла після навантаження. Розроблений пакет вирішує прості задачі набагато швидше ніж будь який з професійних, оскільки він використовує найпростіші методи з точки зору алгоритмічності

складності, а також написаний на швидкій мові програмування Fortran, і не має «тяжкої» графічної оболонки. Розробка цілком може вирішувати прості задачі аналізу твердого деформованого тіла з задовільною точністю и швидкодією.

СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. A. Ya. Grigorenko and S. N. Yaremchenko. Analysis of the stress–strain state of inhomogeneous hollow cylinders. International Applied Mechanics, Vol. 52, No. 4, July, 2016.
2. А. Я. Григоренко, Т. Л. Ефимова, Ю. А. Коротких. Численное решение задачи об осесимметричных свободных колебаниях цилиндра из непрерывно-неоднородного материала на основе сплайн-аппроксимации. Доповіді Національної академії наук України, 2015, № 9.
3. Femap Guide [Електронний ресурс] <https://structures.aero/femap-tutorial/>
4. М.С. Можаровський. Теорія пружності, пластичності і повзучості - Вища шк., 2002. – 308 с.: іл.
5. Пакет ANSYS [Електронний ресурс] <http://www.ansys.com>
6. Огляд FEA-пакетів [Електронний ресурс] <https://sapr.ru/article/6797>
7. Бенчмарки FEA-пакетів [Електронний ресурс] <http://www.nafems.com>
8. Список найбільш поширених комплексів ПЗ на основі ЧМ одного класу з MCE [Електронний ресурс] <http://www.cprsys1.demon.co.uk>
9. Список найбільш поширених безкоштовних FEA-пакетів [Електронний ресурс] <http://www.vtt.fi/rte7/femsivut.htm>
10. Розробники ПЗ Mefisto [Електронний ресурс] <http://www.ann.jussieu.fr>
11. С. К. Голушко, В. В. Горшков, А. В. Юрченко. О двух численных методах решения многоточечных нелинейных краевых задач. Інститут обчислювальних технологій СО РАН, Новосибірськ, Росія, том 7, № 2, 2002.
12. Аналіз чисельних методів [Електронний ресурс] http://stu.sernam.ru/book_dig_m.php?id=1